Laborator si seminar

Programare in Java si software matematic

Adrian Ichimescu

Adrianichimescu@gmail.com

Constructors

What is a constructor?

- initializes an object when it is created
- has the same name as its class
- syntactically is like a method
- have no explicit return type

Java automatically provides a **default constructor** that initializes all member variables to zero. If you define your **own constructor**, the default constructor is no longer used.

Java allows two types of constructors

- No argument Constructors
- Parameterized Constructors

Let's play with constructors

```java
package lab6;

public class NoArgument {
    int num;
    NoArgument(){
        num = 100;
    }

}
```

```java
package lab6;

public class Exercise1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        NoArgument o1 = new NoArgument();
        Parametrized o2 = new Parametrized (10);
        System.out.println(o1.num + " " + o2.num);
    }

}
```

```java
package lab6;

public class Parametrized {
    int num;
    Parametrized (int i){
        num=i;
    }

}
```

Packages

Package is a grouping of related types (classes, interfaces, enumerations and annotations) providing access protection and namespace management.

Packages are divided into two categories:

- Built-in Packages (packages from the Java API)
- User-defined Packages (create your own packages)

Some of the built-in packages in Java are –

- **java.lang** – bundles the fundamental classes (This package is automatically imported)
- **java.io** – classes for input , output functions are bundled in this package
- **java.awt -** Contain classes for implementing the components for graphical user interfaces (like button , ;menus etc)

Why Packages

Packages are used in Java in order

- to prevent naming conflicts
- to control access: protected and default have package level access control. A protected member is accessible by classes in the same package and its subclasses. A default member (without any access specifier) is accessible by classes in the same package only.
- to make searching/locating and usage of classes, interfaces, enumerations and annotations easier, etc.
- To write easier maintainable code

How to create user-defined Packages

Use of the Key word **package**

Package statement is the first line of the source file

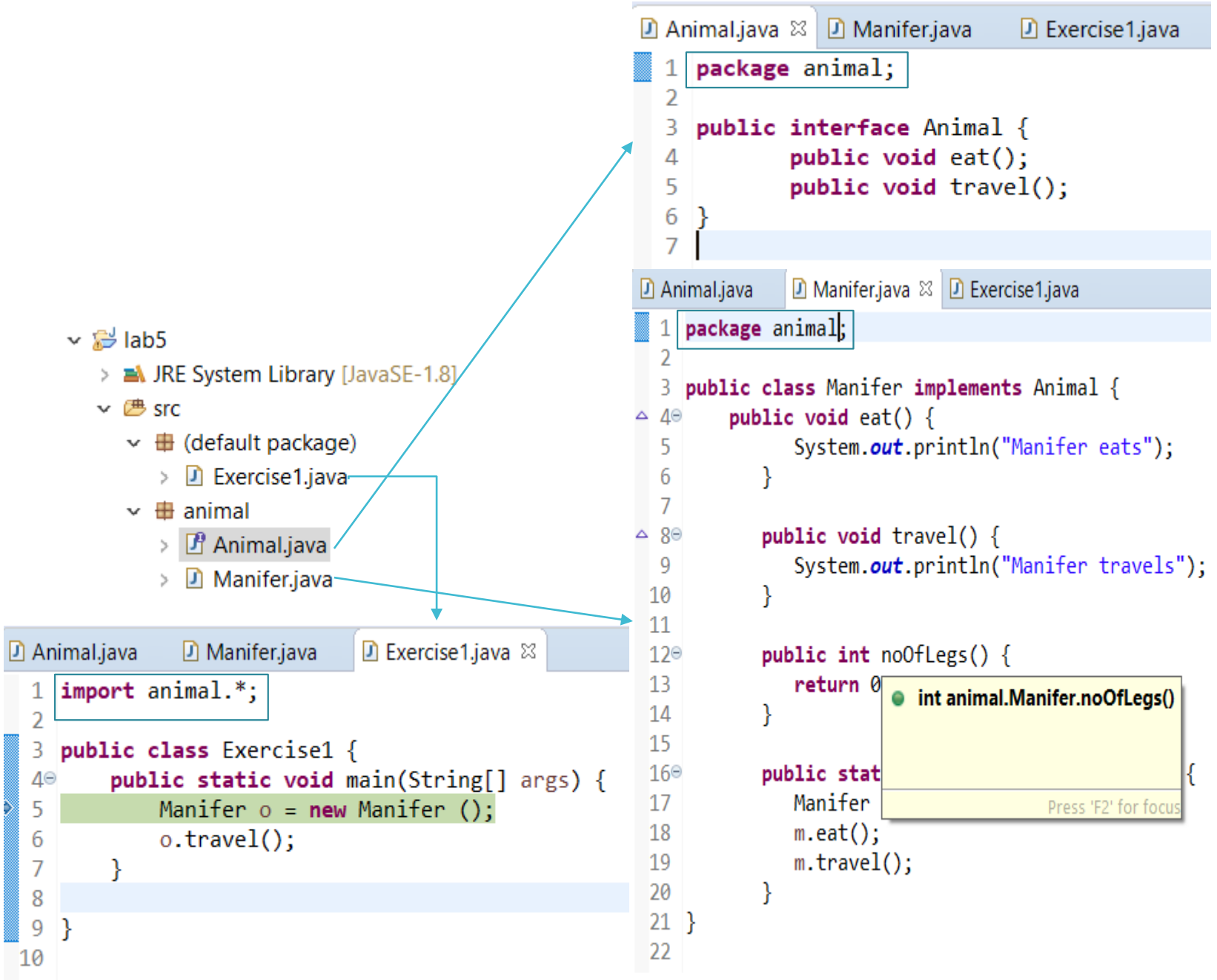
It can be only one package statement in each source file

If **package** statement is not used then the class, interface, enumerations and annotation type will be placed in the current default package

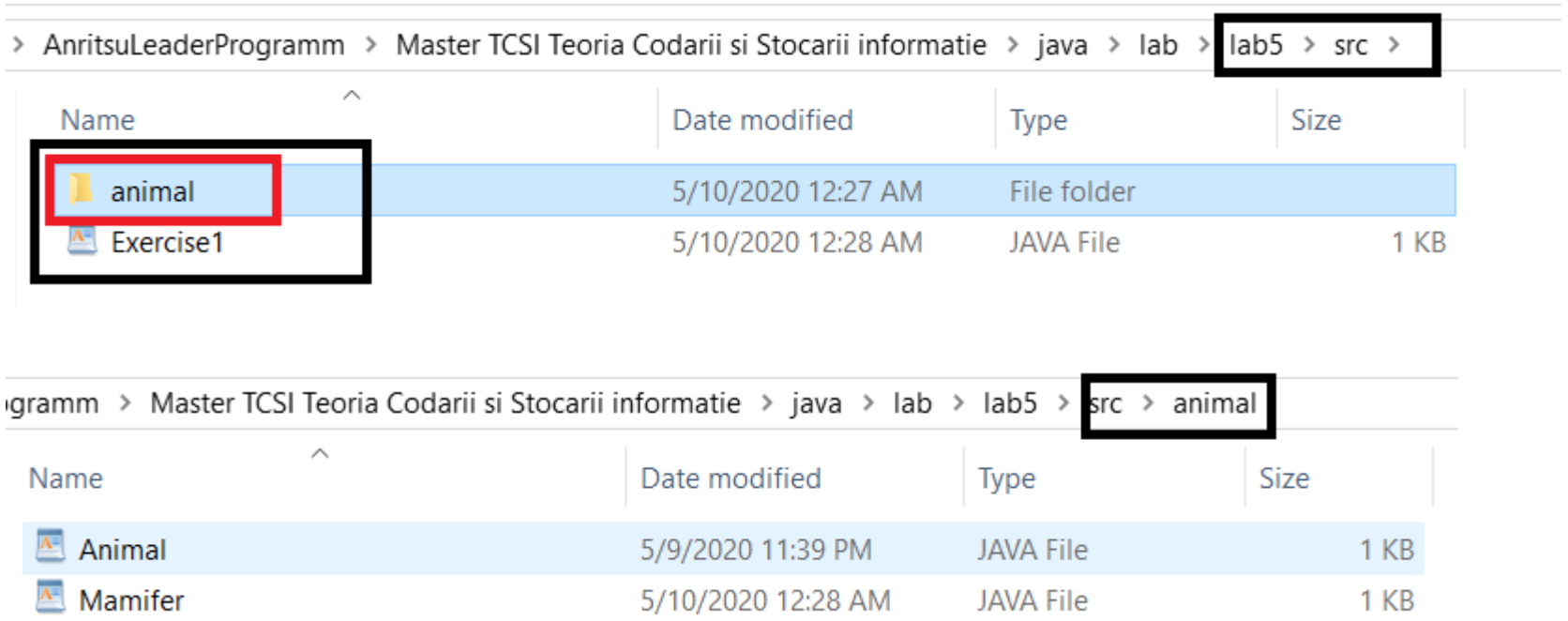
How to use Packages

- If a class wants to use another class in the same package, the package name need not be used. Classes in the same package find each other without any special syntax
- What happens if we want to use a class that is not in the same package? Then use one of the following techniques for referring to a class in a different package.
  - The fully qualified name of the class can be used. For example –payroll.Employee
  - The package can be imported using the **import** keyword and the wild card (*). For example – import payroll.*;
  - The class itself can be imported using the import keyword. For example – import payroll.Employee;
  - Note – A class file can contain any number of import statements. The import statements must appear after the package statement and before the class declaration.

Package

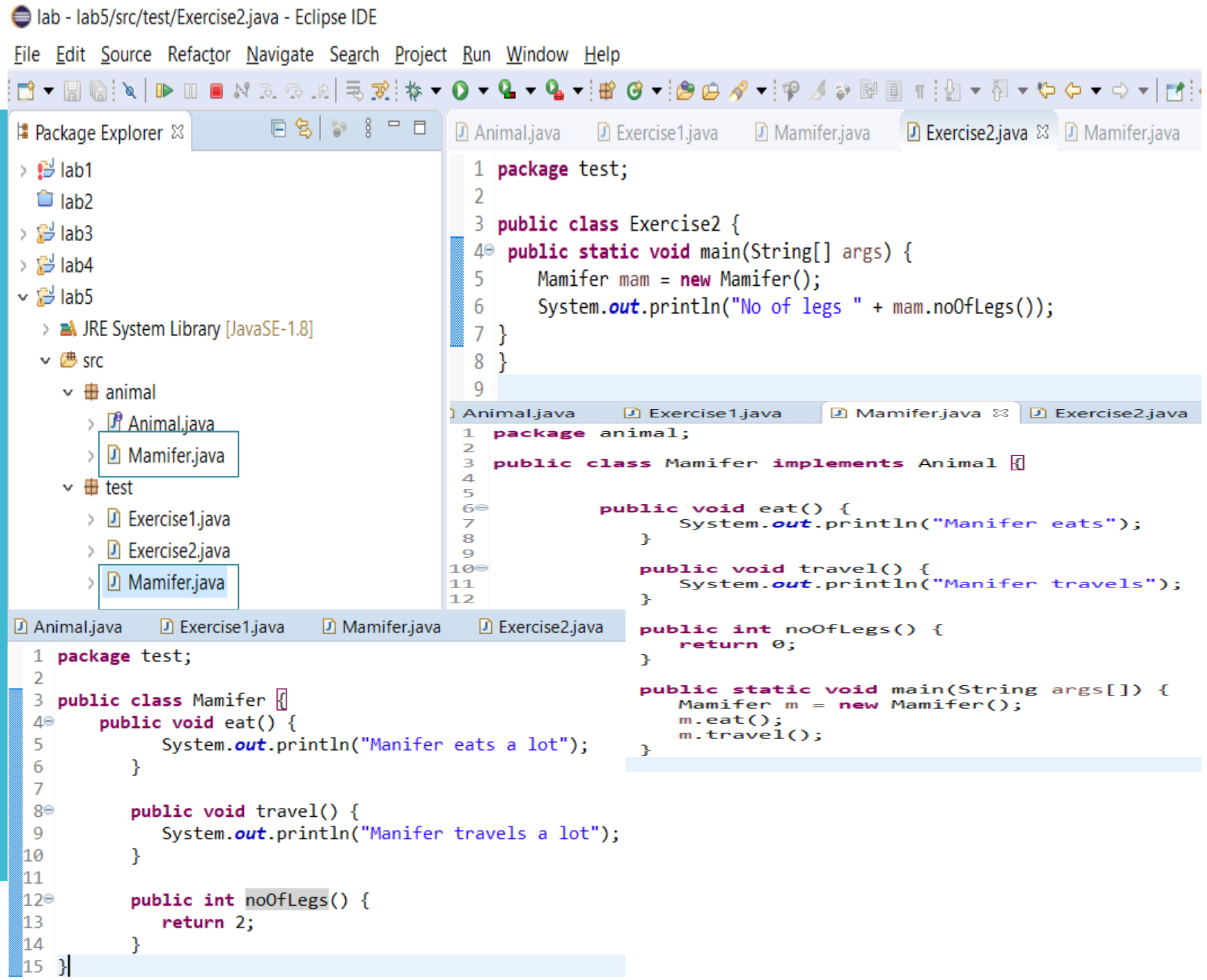

lab5
JRE System Library [JavaSE-1.8]
src
(default package)
Exercise1.java
animal
Animal.java
Manifer.java
Animal.java
Manifer.java
Exercise1.java
1 package animal;
2
3 public interface Animal {
4 public void eat();
5 public void travel();
6 }
7
Animal.java
Manifer.java
Exercise1.java
1 package animal;
2
3 public class Manifer implements Animal {
4 public void eat() {
5 System.out.println("Manifer eats");
6 }
7
8 public void travel() {
9 System.out.println("Manifer travels");
10 }
11
12 public int noOfLegs() {
13 return 0
14 }
15
16 public stat {
17 Manifer
18 m.eat();
19 m.travel();
20 }
21 }
22
int animal.Manifer.noOfLegs()
Press 'F2' for focus
Animal.java
Manifer.java
Exercise1.java
1 import animal.*;
2
3 public class Exercise1 {
4 public static void main(String[] args) {
5 Manifer o = new Manifer ();
6 o.travel();
7 }
8
9 }
10

Package Directory Structure



A package/folder with the name **animals** will be created in the current directory and these class files will be placed in it as shown below.

Packages Prevent Name Conflict



Java I/O Standard Streams

The java.io package contains nearly every class you might ever need to perform input and output (I/O) in Java

All the programming languages provide support for standard I/O where the user's program can take input from a keyboard and then produce an output on the computer screen. Java provides the following three standard streams −

•**Standard Input** − This is used to feed the data to user's program and usually a keyboard is used as standard input stream and represented as **System.in**.

•**Standard Output** − This is used to output the data produced by the user's program and usually a computer screen is used for standard output stream and represented as **System.out**.

•**Standard Error** − This is used to output the error data produced by the user's program and usually a computer screen is used for standard error stream and represented as **System.err**.

System.out is not new for us
Let’s play a bit with **System.in**

Standard Stream System.in

```java
package lab6;

import java.io.*;

public class ReadConsole {

    public static void main(String[] args) throws IOException {
        // Citesc de la tastatura caractere si le afisez pana cand se tasteaza caracterul q
        InputStreamReader cin = null;
        try {
            cin = new InputStreamReader (System.in);
            System.out.println("Enter characters, 'q' to quit.");
            char c;
            do {
                c = (char) cin.read();
                System.out.print(c);
            } while(c != 'q');
          }finally {
              if (cin != null) {
                 cin.close();
              }
          }

    }

}
```

```
Enter characters, 'q' to quit.
aaaaaas
aaaaaas
ssssssdas
ssssssdas
popojjshahsbanskm
popojjshahsbanskm
jhajhjhjhjhjqjaxsjbxja
jhajhjhjhjhjq
```

Homework

Please write a new app in Java that fills an array of chars from the keyboard. The size of the array must be read also from the keyboard.